Abstract

Ant Colony Optimization For **Multicast Routing**

Ying Wang **harrying** Xie

**Institute of Automation, Shanghai Jiaotong University, Shanghai, 200030, China**

[E-mail:wy@controlnet.dhs.org](mailto:wy@controlnet.dhs.org)

very liiiiited amount of memory, and their individual

*Ant* co/ony *ophmization (ACO)* is o nez‹› op//m/z‹z//on *aIgorithm, and has the limitation of stagnation. In this paper, the performance* o/ *ACO* is *improved and the improved ACO is combined u›ith heuristic atgorithm to* solt›e *the* multicast routing problem. Simulation shows *that the results of this atgorithm for multicast routing are better Ihan* flan *heuristic algorithms,’ this algorithm is also u›ell suited for parallel implementation and execution.*

**I. Introduction**

MANY multimedia communication applications require a source to send information to multiple destinations through a communication network. To support these applications, it is necessary to determine a multicast tree of minimal cost for every communication session.

The problem of determining multicast trees is

NP-complete. This problem is also could be considered as the Steiner tree problem. Good heuristic algorithms are of practical. Kyou et al. [I] proposed a heuristic algorithm for Steiner tree problem. This algorithm is based on the MST (minimum Span tree) algorithm, and takes a pseudo poly’nomial time, but this algorithm could not get the best result. Two genetic algorithms were proposed to solve the Steiner tree problem [2] [3]. But these algorithms have some limits, such that the neighbor matrix of the network is taken as the code; it is complex and lowers the efficiency of the algorithm.

Ant Colony Optimization is a new Optimization algorithm; it is used in many Optimization problems now. In this paper, we combined ACO with the heuristic algorithm for multicast routing. This paper is organized as follows: The ACO algorithm is described in section 2. The multicast routing algorithm based on ACO is presented in section 3. Simulation results are given in section 4. Conclusions are presented in section 5.

# II. Ant Colony Optimization

2. 1 Ants in nature

Individual ants are relatively simple insects that have a

**0-7803-6253-5/00/$10.00 ©2000** IEEE.

behavior is apparently random. However, real ants are capable of finding shortest or near shortest paths between a food source and their colony (nest). They lay some pheromone on the ground, thus mark the path by the trail of the substance. An isolated ant moves essentially at random. An ant encountering a previous laid trail can detect it and decide with high probability to follow it, and then reinforce the trail with its own pheromone. The collective behavior, where the more ants following a trail, the more attractive that trail becomes. The process is thus characterized by a positive feedback loop, where the probability with which an ant chooses a path increases with the number of ants that previously chose the same path [4][5][6].

2.2.Ant Colony Optimization Algorithm

According to the behavior of real ants Dorigo et al

[4][5)[6] has defined the heuristic algorithm ACO.

In this section the ACO algorithm in reference [4][51 6] is described by the example of well-known traveling salesman problem. Although the model definition is

influenced by the problem structure, the same approach

can be used to solve other optimization problems.

Given a set of n towns, the TSP can be stated as the

problem of finding a minimal length closed tour that visits each town once. We call *d,d* the length of the path between towns / and ) ; in the case of Euclidean TSP, *d,* 'is the Euclidean distance between / and ,/ (i.e.,



is given by a graph G(/\/, *E)* , where *N* is the set of towns and € is the set of edges between towns (a fully connected graph in the Euclidean TSP). Let

h, (I)(i = 1,2,3 - ii) be the number of ants in town i at

time / and let in = *b,* (/)(/ = 1,2,3 ii) be the total

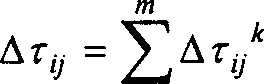


number of ants. Each ant is a simple agent with the following chat-acteristics: it chooses the town to go to with a probability that is a function of the town distance and of the amount of trail present on the connecting edge; to force the ant to make legal tours, transitions to already visited towns are disallowed until a tour is completed (this is controlled by a tabu list); when it completes a tour, it

lays a substance called trail on each edge (i,)) visited.

Let r, (/) be the intensity of trail on edge (i, j) at

time / . Each ant at time / chooses the next town, whet e it will be at time / + 1 . After all the *in* ants coiiiplete its tours the trail intensity is updated according to the following formula:



*k --\*



Where *p* is a coefficient such that A r represents the evapoi-ation of trail between time i and i + u .



the I — **i/z** czziz use *edg@i,* j) izi its tour

*0 otherwise*



Where *Q* is a constant and *L k* ’s the tour length of the *k —th* ant. The coetlicient *p* must be set to a value 0 < *p* < 1 to avoid unlimited accumulation of trail.

At beginning, set the intensity of trail at tune 0, n, (0) to

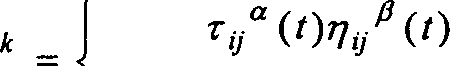
a small positive constant C . In order to satisfy the constraint that an ant visits all the n different towns, each ant is associate with a data structure called the tabu list, that saves the towns already visited up to tune I and forbids the ant to visit them again before a tour have been completed. When a tour is completed, the tabu list is used to compute the ant's current solution (i.e., the distance of the path followed by the ant). The tabu list is then empti d and the ant is free again to choose. */abt/* is defined as the dynamically growing vector which contains the tabu

list of the *k — th* ant. The visibility //,, is the quantity

1 / ‹/, . This quantity is not modified during the run of the ACO, as opposed to the trail, which instead changes

according to the previous formula (1).

The transition probability from town i to town j for the k-th ant is defined as



Since its first development, ACO has been applied to a vat iety of pi oblem areas, only a few of which are mentioned here. These include the traveling salesman problems (TSP) [4][5][6]; static routing (load balancing) in circuit switched telecorrununications networks [7][8].

# III. Multicast routing algorithm for multicast routing

The communication netwoi k is modeled as a graph G(/\/, *E,* C") , where *N* is a set of nodes and *E* is a set o1’ edges. The cost is denoted by the weight C of the edge. A multicast routing group *D ,* is a set of nodes participating in the satire netwoi k activity, and is identified by a unique group address. A multicast routing group tree *T* is a ti-ee spanning all ineiribers of the gi oup, and the cost of the tree is minimum.

Based on ACO algorithm for TSP, we make a few changes for our multicast i outing problem. The multicast i outing probleiri is considered as the Steiner tree problem in this paper. To solve this problems, the Steiner nodes that are not in the multicast routing group must be found.

For the netwoi k G(/\/, €, C) , we fit-st set the ants on the

nodes in the iiiulticast routing group randomly, then chose the Steiner nodes *SD(SD e N I D, D is the mullicst* rotating *group)* with the pi obability y, , when all the ants have finished

choosing the *SD ,* we get the complete gi aph

*.SG(5/\/,* fid, FC’), { *SN —— SD in D, SE ’is* the coiiiparable edge, SC is the comparable cost} for every ants, then find the MST and the total cost L, of MST for it; when the giaph is not connected, a punishment is given to the L,. ACO is used to find the *SD .* for every nodes *n e N / D ,* we set trail intensity r, (u e *N I D] ,* and